

**WHAT IS CLAIMED:**

1 1. A system comprising:

2 a central processing unit operating in response to a  
3 set of instructions for processing information;

4 an interface for providing access to selected circuitry  
5 forming a part of said system on a chip by an external  
6 device; and

7 a set of non-volatile programmable security elements  
8 for selectively enabling and disabling the operation of said  
9 interface to provide a private environment for processing  
10 said information.

1 2. The system of Claim 1 wherein said interface comprises  
2 a JTAG port.

1 3. The system of Claim 1 wherein said interface comprises  
2 an in-circuit emulation port.

1 4. The system of Claim 1 wherein said interface comprises  
2 a port allowing said external device to observe an internal  
3 state of said integrated circuit.

1 5. The system of Claim 1 and further comprising boot  
2 memory for storing security initialization code, said  
3 security initialization code selectively enabled by  
4 programming said set of programmable elements.

1 6. The system of Claim 5 and further comprising boot  
2 memory for storing security initialization code, said  
3 security initialization code selectively enabled by  
4 programming said set of programmable elements.

0946531-00100

7. The system of Claim 1 and further comprising a cache  
associated with said central processing unit, said cache  
including a selected number of lockable entries for storing  
secure information.

1        8. The system of Claim 1 and further comprising a  
2        translation look aside buffer, with said CPU, said  
3        translation look aside buffer including a selected number of  
4        lockable entries for storing addresses to secure information  
5        in memory

1        9. The system of Claim 1 and further comprising on-chip  
2        random access memory including a selected amount of memory  
3        space for storing address translation tables.

1     10. The system of Claim 1 wherein said set of programmable  
2     elements comprises a set of fuses.

1     11. The system of Claim 1 wherein said set of programmable  
2     elements comprise a set of bond options.

1     12. The system of Claim 1 wherein said set of programmable  
2     elements comprises a set of antifuses.

1     13. The system of Claim 1 wherein said set of programmable  
2     elements comprises a set of read-only memory cells.

1     14. The system of Claim 1 wherein said set of programmable  
2     elements comprises a set of write-once memory cells.

1     15. The system of Claim 1 wherein said set of programmable  
2     elements comprises a set of FLASH memory cells.

[illegible]

operating the system in a secure environment in response to the called security procedure when the called security procedure is valid.

[illegible]

enabling the debug circuitry and  
executing boot code pointed-to by the vector.

changing a program counter to the vector such that a fetch of an instruction changing the program counter is completed prior to completion of said step of remapping.

# Qeios ID: 06J07U · V1

1     19. The method of Claim 18 wherein said vector comprises a  
2     CPU reset vector.

20. The method of Claim 16 wherein execution of the security code in boot memory determines that the called security procedure is invalid and said method further comprises the steps of:

```

5      remapping the vector to the boot memory to a location
6      storing second selected security code, the second security
7      code calling a second security procedure;

```

```

8         executing the second selected security code in boot
9         memory to determine if the second security procedure is
0         valid; and

```

1           operating the system on a chip in a secure environment  
2           in response to the second security procedure when the second  
3           security procedure is valid.

21. The method of Claim 16 wherein said step of executing the security code in boot memory to determine whether the called security procedure is valid comprises the substep of searching for the called security procedure in external memory coupled to the system on a chip.

22. The method of Claim 16 and further comprising the step  
of executing default boot code when the called security  
procedure is invalid.

23. The method of Claim 16 wherein said step of determining  
if a security procedure is called for during system  
initialization comprises the substep of reading the state of  
a set of programmable elements.

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1

1 24. The method of Claim 23 wherein said substep of reading  
2 is performed by logic gates.

1 25. The method of Claim 23 wherein said substep of reading  
2 is performed by a central processing unit

001020"ET856460

26. A method of preventing access and observation of  
encached information comprising the steps of:  
generating private information to be encached;  
storing the private information in memory;  
updating a translation look aside buffer with  
descriptors to locations in memory containing the private  
information;  
forcing a cache miss to a selected location in cache to  
be loaded with a selected portion of the private  
information;  
retrieving the selected portion of the private  
information from memory using a corresponding descriptor  
from the translation look aside buffer;  
loading the retrieved portion of the private  
information into the selected location in cache; and  
locking the selected portion of the private information  
in the selected location in cache.

27. The method of Claim 26 and further comprising the step  
of locking the descriptor corresponding to the selected  
portion of the private information in the translation look  
aside buffer.

28. The method of Claim 26 wherein said selected location  
in cache is associated with a replacement counter base and  
said step of locking comprises the substep of resetting the  
replacement counter base to a value higher than the  
replacement counter base associated with the selected  
location in cache.

29. The method of Claim 26 wherein said step of updating the translation look aside buffer comprises the substeps of:

- setting up a translation table including entries for generating the descriptors to memory locations storing the private information;
- updating a replacement counter to point to a current translation look aside buffer entry to be filled;
- forcing a miss to the current translation look aside buffer entry;
- performing a table walk through the translation table to generate a descriptor associated with private information in memory; and
- loading the descriptor obtained from the table walk in the current translation look aside buffer entry.

30. The method of Claim 26 wherein said step of loading the selected portion of the decoded information in cache comprises the step of loading a cache line in instruction cache.

31. The method of Claim 26 wherein said step of loading the selected portion of the private information in cache comprises the step of loading a cache line in data cache.

32. The method of Claim 26 wherein said step of setting up a translation table comprises the step of setting up an emulated translation table.

**0** zero    **1** one    **2** two    **3** three    **4** four    **5** five    **6** six    **7** seven    **8** eight    **9** nine



generating a physical address to the selected one of the memory spaces from the information accessed from the at least one register.

36. The method Claim 33 wherein the at least one register comprises a first register for storing access permissions associated with each of the memory spaces, a second register for storing a cacheability bit associated with each of the memory spaces and a third register for storing a bufferability bit associated with each of the memory spaces.

37. The method of Claim 33 wherein the selected information accessed from the at least one register comprises a base address to at least one second level register controlling access to a selected part of a selected one of the memory spaces and said step of generating a physical address comprises the substeps of:

accessing selected information in the at least one second level register using the base address and a table index from the virtual address; and

generating the physical address from the selected information accessed from the at least one second level register and page index bits from the virtual address.

38. The method of Claim 33 wherein said information includes for each of the memory spaces a pair of access permission bits, a bufferability bit and a cacheability bit.

39. A method of performing an emulated translation table walk comprising the steps of:

emulating a translation register including a plurality of entries populated with descriptors;

emulating an index register storing indices associated with the entries of the emulated translation register;

```
    pointing to the emulated translation register with a
translation base pointer;
```

```

        generating an address including index bits to the
emulated translation register;

```

comparing the index bits from the address with the indices stored in the index register; and

selectively accessing a corresponding descriptor in the translation table in response to said step of comparing.

40. The method of Claim 39 wherein said step of generating an address comprises the step of generating a virtual address forcing a miss to an associated cache.

41. The method of Claim 39 wherein the descriptors comprise selected physical address bits and access permissions and said method further comprises the steps of:

determining from the permissions from the descriptor selectively accessed from the emulated translation table whether a corresponding access to memory is allowed; and

if the access is allowed, generating a physical address to a location in memory using the physical address bits from the accessed descriptor.

in response to said step of comparing, selectively accessing a second level descriptor from the corresponding entry in the second level translation table.

if the access is allowed, generating a physical address to a location in memory using the second level physical address bits from the accessed second level descriptor.

- 1     44. A system comprising:  
2         a central processing unit operating in response to a  
3     set of instructions for processing information;  
4         an interface for providing access to selected circuitry  
5     forming a part of said system on a chip by an external  
6     device; and  
7         a set of programmable security elements for selectively  
8     enabling and disabling the operation of said interface to  
9     provide a private environment for processing said  
10    information.
- 1     45. The system of Claim 44 wherein said central processing  
2     unit, said interface, and said security elements are  
3     fabricated on a single integrated circuit chip.
- 1     46. The system of Claim 45 wherein said integrated circuit  
2     chip further includes on-chip read-only memory.
- 1     47. The system of Claim 45 wherein said integrated circuit  
2     chip further includes on-chip random access memory.
- 1     48. The system of Claim 44 and further comprising memory  
2     storing private code for initializing private operation of  
3     said system.
- 1     49. The system of Claim 44 wherein said system forms a  
2     portion of a hand-held personal appliance.
- 1     50. The system of Claim 49 wherein said hand-held appliance  
2     comprises and audio decoder.

1     51. A hand-held audio decoder comprising:  
2         a central processing unit operating in response to a  
3     set of instructions for decoding a stream of encoded digital  
4     audio data;  
5         memory for storing said set of instructions; and  
6         digital to analog converter circuitry for generating  
7     audio from said decoded stream of digital audio data.

1     52. The audio decoder of Claim 51 wherein said central  
2     processing unit comprises an advanced risk machine.

1        53. The audio decoder of Claim 51 wherein said stream of  
2        encoded digital data comprises a stream of MPEGx, Layer 3  
3        encoded audio data.

1        54. The audio decoder of Claim 51 wherein said stream of  
2        encoded digital data comprises a stream of ACC encoded  
3        digital data.

1        55. The audio decoder of Claim 51 wherein said stream of  
2        encoded digital data comprises a stream of MS Audio encoded  
3        digital data.

1        56. The audio decoder of Claim 51 wherein said decoder is  
2        capable of operating correctly from one AA battery for a  
3        period of at least one hour.

**THE UNIVERSITY OF CHICAGO**

57. A method of synthesized address translation comprising the steps of:

setting up at least one global register having a plurality of entries each for storing access control bits for a corresponding region of memory each comprising a plurality of locations having common access characteristics; and

setting up an individual register for storing a descriptor corresponding to a region of memory having differing access characteristics;

generating an address including an index;

in response to a first state of the index, accessing said descriptor from the individual register; and

in response to a second state of the index, performing the substeps of:

accessing the access control bits from a selected  
one of the global registers pointed-to by said index;  
and

generating a descriptor by merging the access control bits accessed from the selected one of the global registers with selected bits of said address.

58. The method of Claim 57 and further comprising the steps of:

```

    setting up a constant register for storing a constant;
and

```

in response to a third state of the index, accessing a constant from said constant register.

59. The method of Claim 58 wherein the constant register comprises hardwired gates.

1        60. The method of Claim 57 wherein the access control bits  
2        comprise access permission bits, cacheability bits and  
3        bufferability bits.

61. The method of Claim 57 wherein said at least one  
register comprises a first register having a plurality of  
entries each for storing access permission bits for a  
corresponding one of the regions, a second register having  
a plurality of entries each for storing a cacheability bit  
for a corresponding one of the regions and a third register  
having a plurality of entries each for storing a  
bufferability bit for a corresponding one of the regions.



001020"ET85450

1 62. The method of Claim 57 wherein said descriptor  
2 comprises first level descriptor including an second level  
3 index, the method further comprising the steps of:  
4 setting up at least one second level global register  
5 having a plurality of entries each for storing access  
6 control bits for a corresponding region of memory comprising  
7 a plurality of locations having common access  
8 characteristics; and  
9 setting up a second level individual register for  
10 storing a descriptor corresponding to a region of memory  
11 having differing access characteristics;  
12 in response to a first state of the second level index,  
13 accessing the descriptor from the second level individual  
14 register; and  
15 in response to a second state of the second level  
16 index, performing the substeps of:  
17 accessing said access control bits from a selected  
18 one of the second level global registers pointed-to by  
19 said second level index; and  
20 generating a second descriptor by merging the  
21 access control bits accessed from the selected one of  
22 the second level global registers with selected bits of  
23 the address.

1 63. The method of Claim 62 and further comprising the  
2 steps of:  
3 setting up a second level constant register for storing  
4 a second level constant; and  
5 in response to a third state of the second level index,  
6 accessing a second level constant from the second level  
7 constant register.

1 64. The method of Claim 58 wherein the second constant  
2 register comprises hardwired gates.

1 65. The method of Claim 57 wherein the access control bits  
2 comprise access permission bits, cacheability bits and  
3 bufferability bits.

004953.020100  
004953.020100